WHAT IS CLAIMED IS:

1. A computer-based method for determining the optimum join sequence for processing a query having a plurality of tables from a relational database stored in an electronic storage device having a database management system, the method comprising the steps of:

(a) a first pass for determining an optimum join sequence for joining the plurality of tables from the query; and

(b) a second pass for using the optimum join sequence for creating a lowest cost access path plan for processing the query.

2. The method according to claim 1, wherein the first pass performing successive steps until creation of a simulated composite table having all tables from the query, wherein each said step:

creating a set of miniplans for simulating all possible joins of a predetermined subset of the query tables; and

using a cost model calculations for estimating and saving the least expensive join from said set of joins, thereby determining the optimum join sequence.

3. The method according to claim 2, wherein the first pass for each said miniplan storing a used table index, join method, and sorting data, and for each said least expensive join storing names of joined tables, join cost and possible row orderings.

4. The method according to claim 3, wherein the first pass only storing non-redundant miniplan data, and saving partial results of the cost model calculations for future reuse.

5. The method according to claim 1, wherein the second pass performing successive steps until creation of a simulated composite table having all tables from the query, wherein each said step being performed in the optimum join sequence.

6. The method according to claim 1, wherein the query being a SQL query.

7. A computer-based processor system for determining the optimum join sequence for processing a query having a plurality of tables from a relational database stored in an electronic storage device having a database management system, the system comprising:

means for performing a first pass for determining an optimum join sequence for joining the plurality of tables from the query; and

means for performing a second pass for using the optimum join sequence for creating a lowest cost access path plan for processing the query.

8. The system according to claim 7, wherein the first pass means performing successive steps until creation of a simulated composite table having all tables from the query, wherein each said step:

creating a set of miniplans for simulating all possible joins of a predetermined subset of the query tables; and

using a cost model calculations for estimating and saving the least expensive join from said set of joins, thereby determining the optimum join sequence.

9. The system according to claim 8, wherein the first pass means for each said miniplan storing a used table index, join method, and sorting data, and for each said least expensive join storing names of joined tables, join cost and possible row orderings.

10. The system according to claim 9, wherein the first pass means only storing non-redundant miniplan data, and saving partial results of the cost model calculations for future reuse.

11. The system according to claim 7, wherein the second pass means performing successive steps until creation of a simulated composite table having all tables from the query, wherein each said step being performed in the optimum join sequence.

12. The system according to claim 7, wherein the query being a SQL query.

13. A computer usable medium tangibly embodying a program of instructions executable by the computer to perform a computer-based method for determining the optimum

join sequence for processing a query having a plurality of tables from a relational database stored in an electronic storage device having a database management system, the method comprising the steps of:

(a) a first pass for determining an optimum join sequence for joining the plurality of tables from the query; and

(b) a second pass for using the optimum join sequence for creating a lowest cost access path plan for processing the query.

14. The method according to claim 13, wherein the first pass performing successive steps until creation of a simulated composite table having all tables from the query, wherein each said step:

creating a set of miniplans for simulating all possible joins of a predetermined subset of the query tables; and

using a cost model calculations for estimating and saving the least expensive join from said set of joins, thereby determining the optimum join sequence.

15. The method according to claim 14, wherein the first pass for each said miniplan storing a used table index, join method, and sorting data, and for each said least expensive join storing names of joined tables, join cost and possible row orderings.

16. The method according to claim 15, wherein the first pass only storing non-redundant miniplan data, and saving partial results of the cost model calculations for future reuse.

17. The method according to claim 13, wherein the second pass performing successive steps until creation of a simulated composite table having all tables from the query, wherein each said step being performed in the optimum join sequence.

18. The method according to claim 13, wherein the query being a SQL query.